DESIGN OF 30-TAP FIR FILTER USING VHDL

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGRRE OF

Bachelor of Technology

in

Electronics And Instrumentation Engineering

By

ACHINTA ROY

(Roll No. 110EI0144)

Department of Electronics And Communication Engineering
National Institute of Technology
Rourkela-769008

NATIONAL INSTITUTE OF TECHNOLOGY

CERTIFICATE

This is to certify that the thesis titled **"Design of 30-Tap FIR filter in VHDL "** submitted by **Achinta Roy (Roll no. 110EI0144)** in partial fulfillment of the requirements for the award of **BACHELOR OF TECHNOLOGY** in **ELECTRONICS AND INSTRUMENTATION ENGINEERING** at **National Institute of Technology, Rourkela** is an original work carried out by him under my supervision and guidance.

The matter embodied in the thesis has not been submitted to any University/ Institute for the award of any Degree.

Date: 12/05/2014

Prof. Ayas Kanta Swain

Assistant Professor

Department of Electronics And Communication

National Institute of Technology, Rourkela

ACKNOWLEDGEMENT

I take this opportunity to express my gratitude and deep regards to my guide Prof. A.K Swain, Assistant Professor, Department of Electronics And Instrumentaion Engineering, for his guidance, monitoring and constant encouragement throughout the course of this project. His invaluable guidance and immense help are embodied in this dissertation.

I also express my sincere gratitude to Prof. S.K Meher, Head of Department, Department of Electronics And Communication Engineering for their keen interest and unfailing inspiration throughout the course of the project.

I am thankful to other M.Tech and research scholars of Department of Electronics And Communication Engineering for providing all kinds of support in the lab and helping me throughout the work.

I am also grateful to the institute's laboratory and other facilities for providing me with required resources for the completion of this project.

Finally, I thank all those who are involved, directly or indirectly, throughout the course of the project.

Achinta Roy

110EI0144

List of Figures

List of Tables

ABSTRACT

A filter may be required to have a given frequency response, or a specific response to an impulse, step, or ramp, or simulate an analog system. Depending on the response of the system, digital filters can be classified into Finite Impulse Response (FIR) filters & Infinite Impulse Response (IIR) filters. The thesis deals with design of generic 30-tap FIR filter on FPGA. The thesis is focused on Design structure and occupied silicon space, needed for implementation of filter in FPGA. The results are IP macros of simple FIR filter that are full configurable using generic parameters.

Both macros were verified in a verification environment which consists of test blocks (VHDL) and a comparative model (Matlab). A design of generic FIR filter is described in this work. Next there are described final designs of the IP macros, results and process of the verification, implementation and gate-level verification.

CONTENTS

Pages

1. Introduction

1.1 Filters:

The basic operation in digital signal processing is filtering. This operation is widely used in many electronic devices to cancel part of signal that is redundant or damages the signal.

Filters have two uses: signal *separation* and signal *restoration*. Signals which are corrupted by interference and noise require separation techniques. A device for measuring the electrical activity of a baby's heart inside the mother's womb will be corrupted by breath signal and heartbeat signal of the mother. At such times filters are used to separate the signals and analyse them individually..

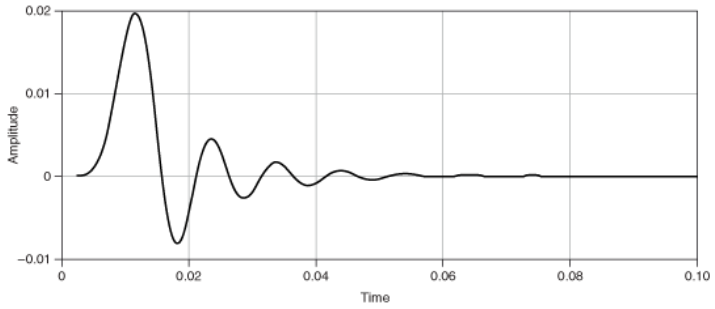
When signal gets distorted the process of signal restoration is used. Audio recording made with poor equipment is filtered to give better sound signal output than the original it previously produced..
They can be either solved by analog or digital filters.

Analog filters, cheaper, faster, and large dynamic range in both amplitude and frequency. Digital filters in comparison, vastly superior performance level that can be achieved. Quality is better than analog filters to digital filters can achieve performance unique. The filtering problem is approached makes a dramatic difference. With analog filters, emphasizing precision and stability, such as resistors and capacitors in electronics, controls have limitations. In comparison, digital filters are often ignored in order to better filter performance. The emphasis shifts signal constraints, and the processing of theoretical issues.

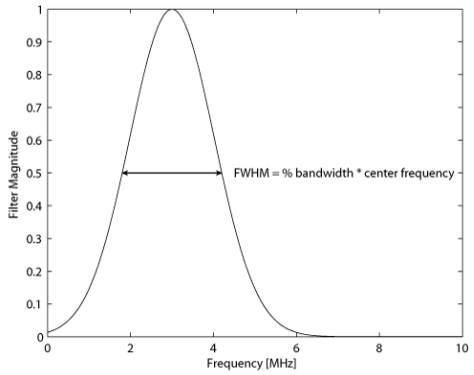
It is a filter in the time domain of the input and output signals are telling the DSP. Because it is usually created by the signs of the time pattern at regular intervals. But this model is not the only way to open. The second most common way that the space is equal to the sample period. Many other domains are possible; however, time and space are by far the most common.

Each linear filter impulse response, a step response and frequency response. Full information about each of these responses is the filter, but in a different form. One of the three specified, the other two are fixed

and can be calculated directly. They react in different situations to describe the filter because the representations of the three, the most important.



Impulse Response



Frequency Response

The easiest way to implement a digital filter response is convolution of the digital impulse response with the discrete time signal input. All linear filters can be made in this manner. Filter kernel: impulse response used in this way, the filter designers give a special name .

There is also another way to do that is the principle of the digital filter. By applying a filter, convolution product samples, weighting each sample in the input, and is calculated by adding them together. Recursive filters, along with the points from the input, output, using the previously calculated values , these representatives. Instead of using a filter kernel, a set of coefficients derived from the principle of recursive filters. Feeding the output of the filter is a recursive filter gives inspiration. Impulse responses are sinusoidal with exponentially decaying amplitude. Therefore, they have infinite long impulse

responses. IIR filters or infinite impulse response filters, so -called recursive filters. The process of convolution filters, Finite Impulse Response or FIR filters, which are observed in the course of.

When the output of the system is inspired by the input impulse response. Input (also called an edge, and an edge response) is a step in the same manner, the step response is the output. Comprehensive induction step, so that the impulse response is an important step in the response. (1) The filter is cleared by a step in the food and see the waveform, or (2) to integrate the impulse response: This step provides two ways to find a response. (Mathematically correct codes to be used for the integration of continuous, discrete integration, i.e., a running total, while the discrete codes are used). Frequency response impulse response (FFT algorithm using the method) DFT can be seen taking

1.2 Representation of information in signals:

The most important part of any DSP task is to understand how the information in the signal. There are many ways that information can be a signal. This is especially true if the signal man. AM, FM, single-sideband, pulse modulation, pulse width modulation, etc. The list goes on and on: For example, consider all that devised modulation. Fortunately, the nature of information that is common to represent the signals is only two ways. Information is represented in the time domain, frequency domain and the information referred to in this call.

When the time domain, as is any information that describes what is occurring range. For example, imagine an experiment to study the sun light output. Light output is measured and recorded every second. The signal for each sample instant, and event status indicates what is going on. If a solar flare occurs, the signal of each sample directly, without reference to any other kind of information, etc. It is understood that the information on the time, duration, growth over time, offers. This is the easiest way to provide information in a signal.

Instead, information is represented in the frequency domain becomes more indirect. Many things in our universe show periodic motion. For example, hitting a wine glass with a finger produces a ringing sound, vibrate; A grandfather clock pendulum back and forth match; their axis of rotation of the stars and the planets back and forth and around each other, and so on. The term operating frequency, phase, and amplitude measures, with information about the production system can run more often. We want to sample the sound produced by the ringing of wine glasses. And harmonics of the fundamental frequency of vibration of the mass and elasticity of the material related. A model, itself, does not have any information about run -time, and a glass of wine so we have no information. The relationship between the information signals is spotty.

This measure is of great importance and frequency responses. Describes how to edit the information system as the time domain response. In contrast, the frequency response in the frequency domain shows that as the information has changed. Both applications can improve it with a filter, filter design, because this distinction is absolutely critical. Frequency domain and time domain results in a poor performance, and vice versa

Recursive filters without having to make a long loop, is a great way to achieve a long impulse response. They run very quickly, but other than digital filters have lower performance and flexibility. Exponential impulses responses are disintegrating because of their recursive filters, Infinite Impulse Response (R) are called filters. The (FIR) filters, also known as Finite Impulse Response, material, to distinguish them from the spirit of the digital filters.

1.3 Advantages of FIR over IIR:

Taking a look into the design of FIR filters we see its advantages and disadvantages. IIR design can be done with certainty because of the presence of the analytical base and closed form. In FIR there is a lot of uncertainty. And at every step you are not sure whether you are proceeding in the correct manner and whether the specs are satisfied so it has a blind start. In IIR there were analytical formulas for calculating the Butterworth order and Chebyshev order. There is nothing like this in FIR. In FIR you only have empirical formulas which may or may not work. If empirical formulas give the order of 17 you might have to fix an order of 20 or 21. Empirical formulas always have this kind of tolerance and uncertainty. But amidst that many drawbacks what is the real need to use
FIR?

It is because FIR is unconditionally stable and has a linear phase. Linear phase is a strict requirement. For example, in data processing if a rectangular pall becomes near because of delay distortion then rectangular palls does not convey what you wish to convey. In speech processing linear phase is a strict requirement so you have to use it. There are two advantages: one is, it is linear phase and the other is that it is unconditionally stable. There is a third advantage. If you have a non-causal FIR then you can make it causal by simply shifting the whole thing to the right and then you multiply with the required number of delays. So realizability of FIR is not a problem and is not a great advantage but this is one of them. And the disadvantage is that you have to use for the same specs a large order. So the cost goes high. What can be done by the 2nd order may require a 20th or 30th order FIR so the cost goes up. And if you implement it by convolution in the time domain it is a very slow process. If you write the software or do the required hardware multiplication, addition and so on it is a slow process. Nevertheless this is not a disadvantage because convolution can always be calculated by DFT. If h(n) and x(n) you have to convolve you take H(DFT)k and X(k) and multiply the two and take the inverse DFT. DFT is also a slow process and that is why FFT is much popular

1.4 Terms used to describe FIR filter:

- Impulse Response - The "impulse response" of a FIR filter is the set of FIR coefficients. (If you put an "impulse" into a FIR filter consisting of a "1" and then many samples of "0" samples, the output of the filter will be the set of coefficients, as the 1 sample moves past each coefficient in turn to form the output.)
- Tap - A FIR “Pipeline” is simply a coefficient / delay is set. (The "N" as designated) FIR The number of taps , a sign of 1) the amount of memory required to implement the filter , the necessary calculations, 2) number , and the filter " filter " 3) size ; Effect , such as pipes and stop -band attenuation and low ripple , narrower filters , meaning
- Multiply-Accumulate (MAC) - An FIR In the context of a "Mac" as a result of amplification of the corresponding coefficient to collect sample data delayed action. The first information report usually requires a Mac with one tap. A majority of microprocessors, DSP instruction cycle MAC, enabling operation.
- Transition Band – Frequencies existing between band pass and band stop. More numbers of taps will be required for filter implementation if transition band is narrow.
- Delay Line - FIR The calculation of "z ^ -1" delay implementing elements set of memory elements.
- Circular Buffer - Because it occurs around the beginning of the end of incrementing or decrementing from the beginning to the end of the round because it makes the “circular “This is a special buffer. Without having to move the data in non-volatile memory in a circular FIR Models with delay line “movement " is presented by the implementation of the DSP microprocessors . When a new sample buffer is added , it will automatically replace the oldest one

1.5 FIR Filter:

"FIR" means "Finite Impulse Response". If we put in an impulse, that is, a single "1" sample followed by many "0" samples, zeroes will come out after the "1" sample has made its way through the delay line of the filter.

In the common case, the impulse response is finite because there is no feedback in the FIR. A lack of feedback guarantees that the impulse response will be finite. Therefore, the term "finite impulse response" is nearly synonymous with "no feedback".
However, if feedback is employed yet the impulse response is finite, the filter still is a FIR. An example is the moving average filter, in which the nth prior sample is subtracted (fed back) each time a new sample comes in. This filter has a finite impulse response even though it uses feedback: after N samples of an impulse, the output will always be zero.

The difference function of FIR filter that defines how the input signal is related to the output signal is:

$y[n] = x[n]b[0] + x[n-1]b[1] + x[n-2]b[2] + \ldots\ldots\ldots\ldots\ldots\ldots + x[n-o-1]b[o-1]$ (1)

where b[i] are coefficients of the filter, x[n] is an input signal, y[n] is an output signal and 'o' is the order of the filter. The transfer function of a FIR filter is:

$$H(z) = \sum_{n=0}^{o-1} b[n] \cdot z^{-n} \tag{2}$$

The above equation is the filter's equation in z domain where b[n] represents the filter co-efficient also called as the filter response. The output of a filter to an input response of x[n] is determined by the convolution function

$$y[n] = h[n] * x[n] \tag{3}$$

The Lth-order LTI FIR filter is graphically interpreted in Fig . It can be seen to consist of a collection of a "tapped delay line," adders, and multipliers. One of the operands presented to each multiplier is an FIR coefficient, often referred to as a "tap weight" for obvious reasons. Historically, the FIR filter is also known by the name "transversal filter," suggesting its "tapped delay line" structure.

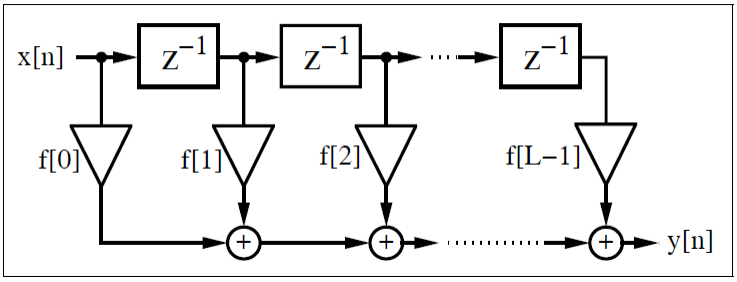


Figure 1: Direct form of FIR filter.

FIR Filter with transposed structure

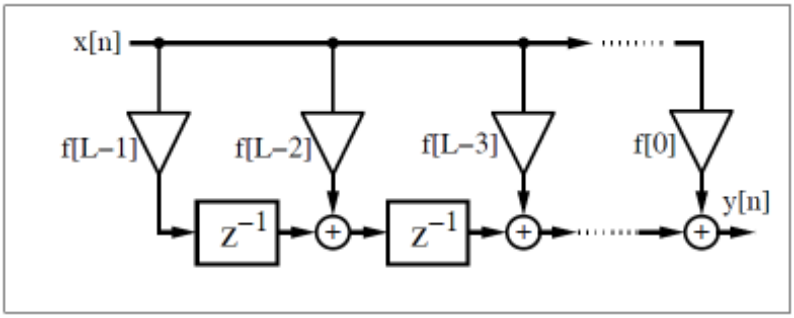


Figure 2: Transposed form of FIR filter.

A variation of the direct FIR model is called the transposed FIR filter. It can be constructed from the FIR filter in Fig. 10.1 by:

– Exchanging the input and output

– Inverting the direction of signal flow

– Substituting an adder by a fork, and vice versa

A transposed FIR filter is shown in Fig. 2 and is, in general, the preferred implementation of an FIR filter.

The benefit of this filter is that we do not need an extra shift register for x[n], and there is no need for an extra pipeline stage for the adder (tree) of the products to achieve high throughput.

The next section shows the description of the design of the above generic filter in VHDL language in Xilinx environment. The whole process of the design is writing the VHDL code of the description design, simulation of the design and verification of the Filter model by writing a test bench for it and then implementing the model in FPGA.

2. Experimental

2.1 General Description

The designed FIR filter implements the function which is written in eqn 2. The architecture of the filter is fully sequential using one time-multiplexed multiplier and one adder. The outputs of filter are registered. Coefficients b are stored in the internal register array and can be read/written using buses u_pipe and y_pipe.

2.2 Features

1. Input bus width u (signal) is Nx, generic parameter.
2. Output bus width y (signal) is log2o+Ny, both generic parameters.
3. Coefficient bus widths ‘u_pipe’ and ‘y_pipe’ are Nc, generic parameters.
4. Coefficients can be loaded in a serial or parallel way using address bus ‘b’.
5. The addressing of coefficients is described below.
6. Order of filter is specified by generic parameter ‘o’. Besides this, ‘log2o’ generic parameter must be specified to determine the width of data path.
7. Internal calculations are executed in full precision. The widths of internal buses are Nx+Nc+log2o.
8. Computation starts with high level of the edge triggered clock signal. The finishing of the computation is indicated by high level of the edge triggered clock signal

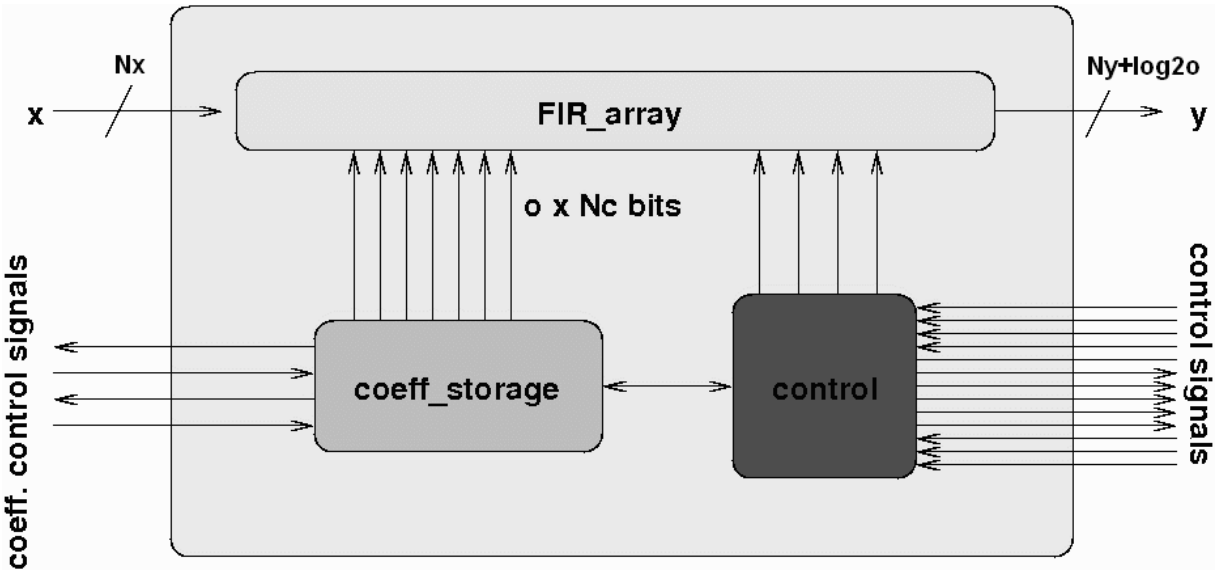


Figure 3: Top level schema of the FIR filter.

2.3 Interface

Generic	Type	Description
o	natural	The order of the filter
log2o	natural	Used for data path width – saturation margin
Nx	natural	Input data word width
Nc	natural	Coefficient data word width
Ny	natural	Output data word width

Table 1

Signal	Direction	Type	Description
u	in	signed(width-1 downto 0)	Data Input
y	out	signed(width-1 downto 0)	Data Output
reset	in	std_ulogic	Global reset. Active low
clk	in	std_ulogic	Main clock signal. Rising edge is the active one.
b	generic	signed(width-1 downto 0)	Co-efficient vector
y0	generic	signed(width-1 downto 0)	Final output pipeline register
u_pipe	generic	signed_vector(‘brange)	Acts as the shifting pipeline of the co-efficient vector
y_pipe	generic	signed_vector(‘brange)	Acts as the adder and multiply multiplexer

Table 2

2.4 Coefficent Addressing

Address Bus	Mapped Co-eeficent
FFEF	b[0]
FFED	b[1]
FFE8	b[2]
FFE6	b[3]

Table 3

Sequential access allows serial loading of coefficients. Each register for coefficient is connected to the shift register and data are written from input coefficient bus *u_pipe*. Sequential mode is activated by high level of signal *clk*. With rising edge of clock, coefficients are shifted to the right and value is written to the left side of shift register (*b [0]*). This mode allows for example computation of autocorrelation function.

2.5 Design Summary:

fir Project Status			
Project File:	fir.xise	**Parser Errors:**	No Errors
Module Name:	fir	**Implementation State:**	Synthesized
Target Device:	xc5vlx20t-2ff323	**• Errors:**	No Errors
Product Version:	ISE 14.2	**• Warnings:**	4 Warnings (0 new)
Design Goal:	Balanced	**• Routing Results:**	
Design Strategy:	Xilinx Default (unlocked)	**• Timing Constraints:**	
Environment:	System Settings	**• Final Timing Score:**	

Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Registers	16	12480	0%
Number of Slice LUTs	218	12480	1%
Number of fully used LUT-FF pairs	8	226	3%
Number of bonded IOBs	34	172	19%
Number of BUFG/BUFGCTRLs	1	32	3%
Number of DSP48Es	27	24	112%

Figure 4: Xilinx 14.2 report of the filter design

2.6 RTL Description:

Top Level Schema

Figure.5 and Figure.6 show the top level description of the circuit. Top level circuit consists of three blocks. Adder:20 and Multiplexer(adder and multiplier) in Figure.5 and dflip-flop and Multiplexer(adder and multiplier) in the Figure.6

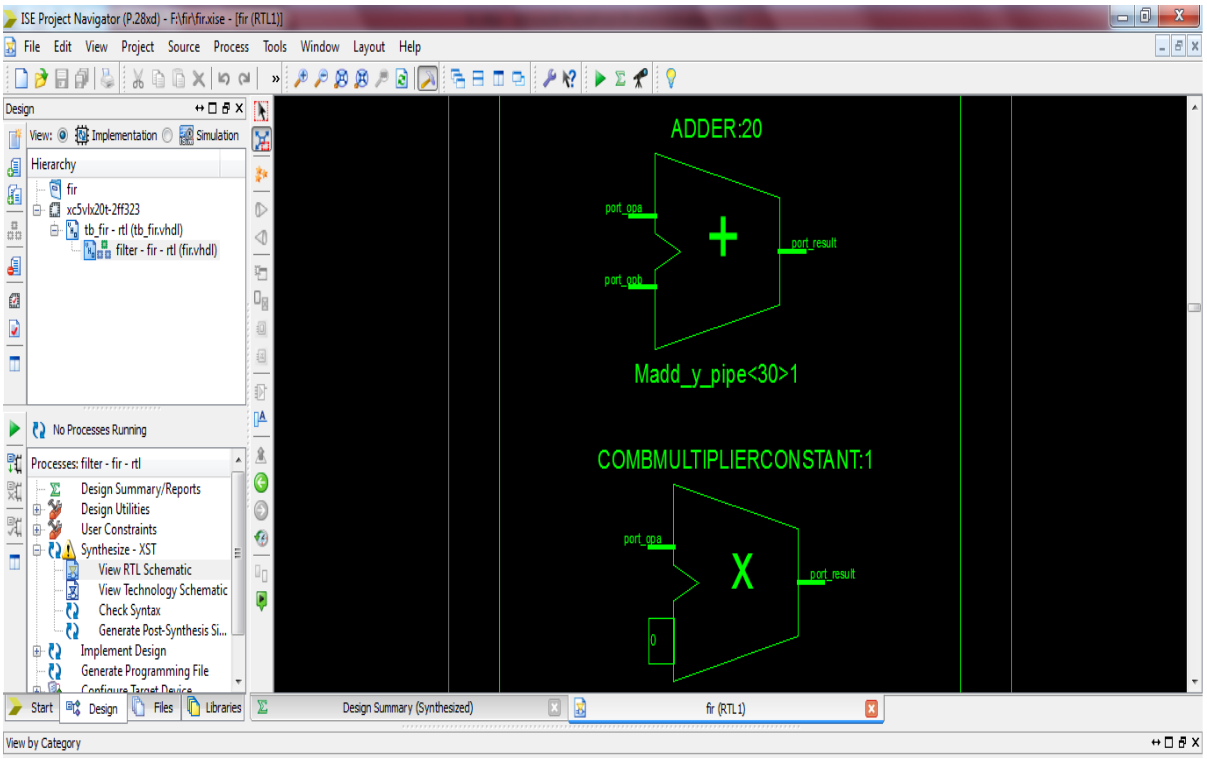


Figure 5: VHDL Top level RTL Schema1

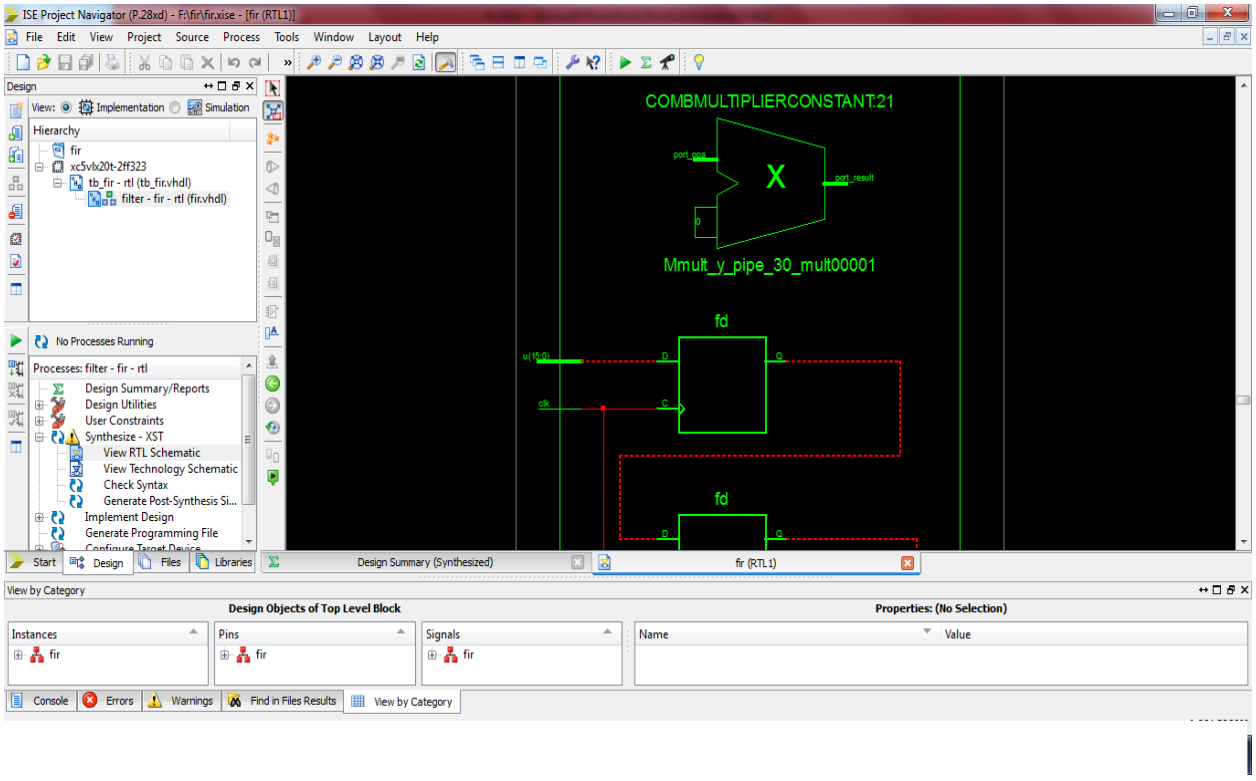


Figure 6: VHDL Top level RTL Schema2

Full Schematic

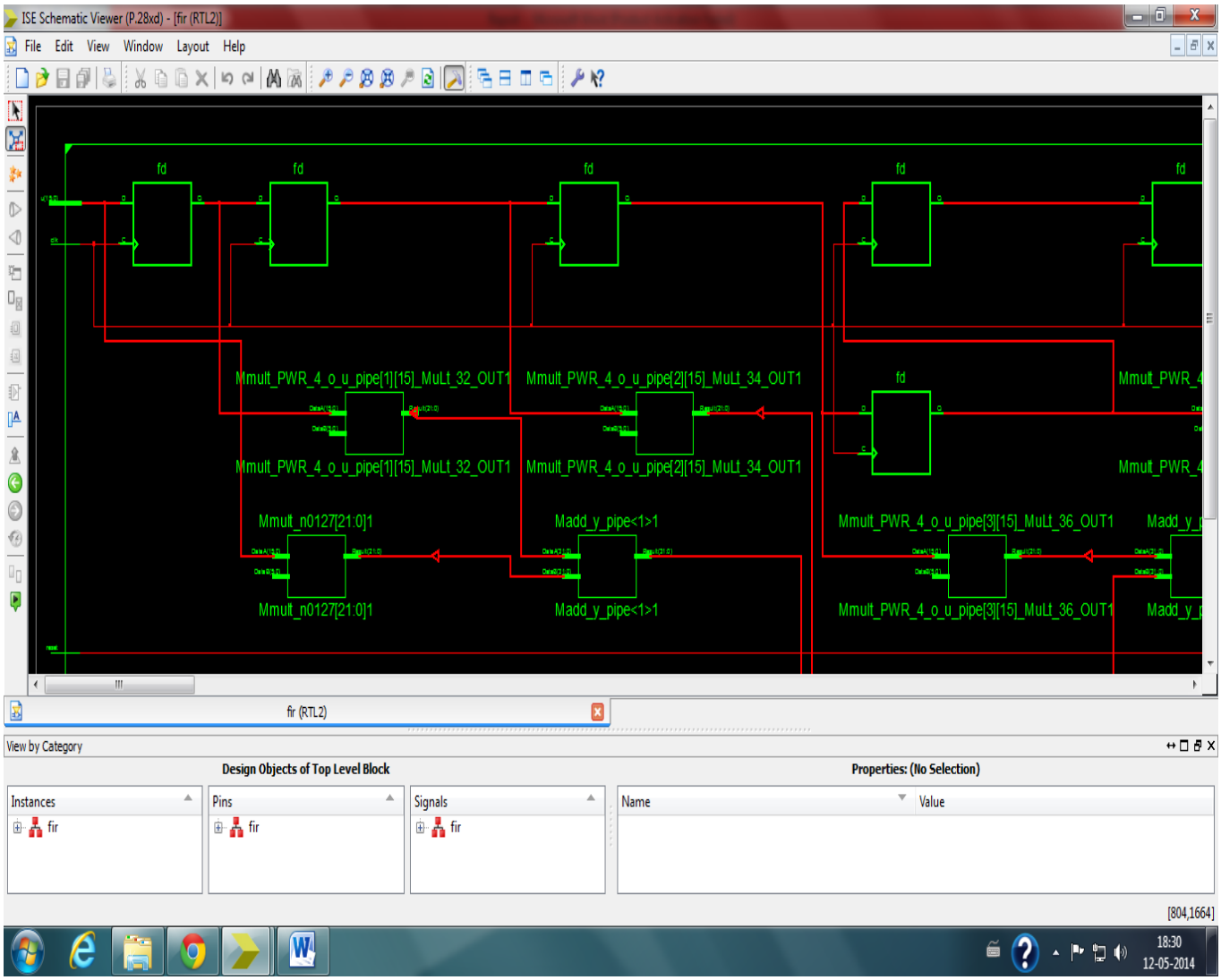


Figure 7: Xilinx 14.2 full schematic of FIR order 30 filter.

2.7 Test Bench Verification:

Signal *zero* is set after to all pipeline registers at the input of the filter are written values. Then, by rising edge of clock, pipeline registers and also output and input registers are reset. The waveform of the testbench is in Figure 6.

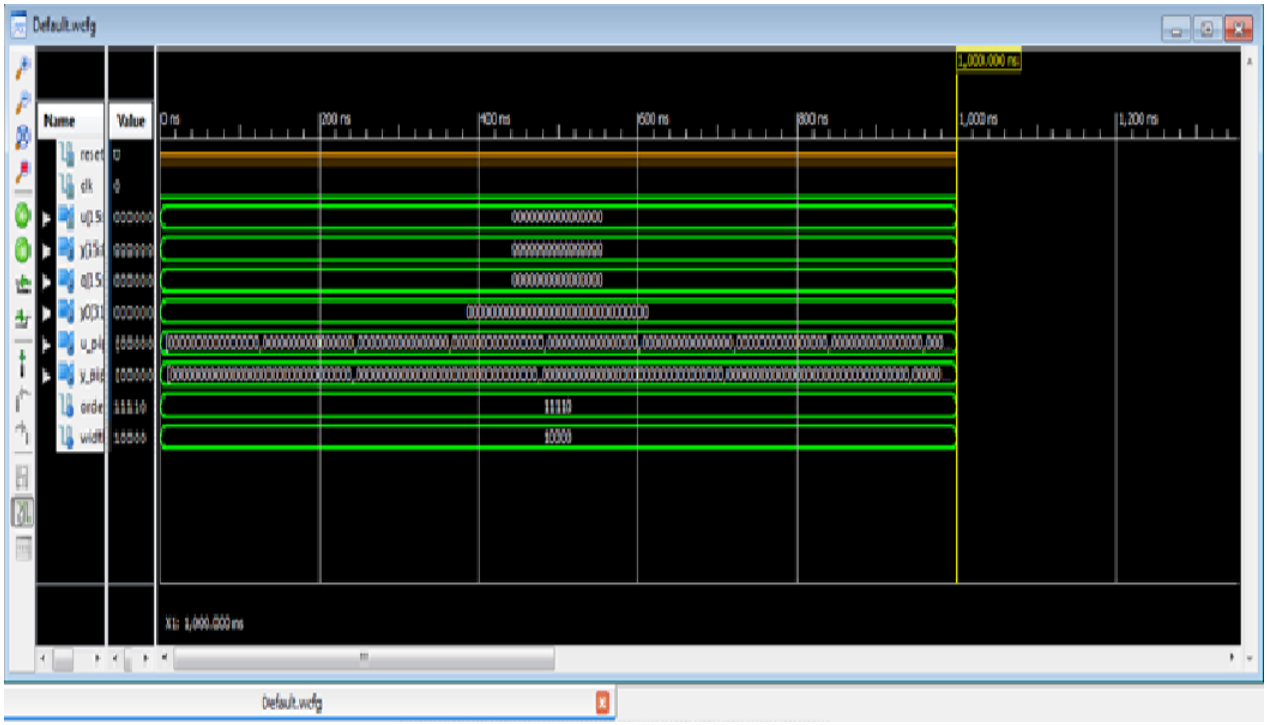


Figure 8 : Testbench simulation of 30 order FIR flter

3. Conclusions

Conclusions:

Design of the generic 30-tap FIR in Xilinx 14.2 Ver was successful. It was verified in the standard verification environment of the FPGA laboratory and are ready to use. The design is suitable for applications that require low power consumption and a small occupied silicon space. The main advantage of the design is that they can be implemented in whichever FPGA; meaning that they are not dependent on the platform.

The theme of the next work could be the design of the interpolated FIR filters with several interpolation filters that could further reduce the order of the designed filter.

References:

[1] J.Douša, VHDL Language, Textbook CTU FEE 2003, in Czech

[2] T.W. Parks and C.S. Burrus, Digital Filter Design. New York:Wiley,1987

[3] Richard S. Juszkiewicz, An Analysis of Interpolated Finite Impulse Response Filters and Their Improvements, IEEE Signal Processing Magazine, November 2005

[4]Xilinx FPGA datasheets, www.xilinx.com

[5]Digital Signal Processing by Prof Dutta Ray, IIT Kharagpur, www.nptel.ac.im

[6] J.G. Proakis and D.G. Manolakis, Digital Signal Processing-Principles,Algorithms and Applications New Delhi: Prentice-Hall, 2000

[5]Matlab 10.1 Product Help